

What is claimed is:

1. A method for managing object to relational one-to-many mapping for an object model mapped to a relational database, the method comprising steps of:
obtaining, for a source object having a primary key value and being manipulated in a corresponding source table of the relational database, mapping meta-data including information of a corresponding target table for storing at least one target object with which the source object has a one-to-many relationship of privately owned type and information of a foreign key of the corresponding target table;
generating an instruction to manipulate the at least one target object in the corresponding target table based on the mapping meta-data; and
manipulating the at least one target object in the database by executing the instruction on the database.
2. The method as claimed in claim 1, wherein the instruction generating step generates an instruction to insert, read, delete or update the each target object in the database.
3. The method as claimed in claim 1, wherein each target object in the object model does not contain information regarding the one-to-many relationship.
4. The method as claimed in claim 1, wherein the generating step and the manipulating step are performed on each target object with which the source object has the one-to-many relationship of the privately owned type..
5. The method as claimed in claim 1, wherein the obtaining step is carried out when a system performing the method is initialized.
6. A method for managing object to relational one-to-many mapping for an object model mapped to a relational database, the method comprising steps of:
obtaining, for a source object having a primary key value and being inserted in a corresponding source table of the relational database, mapping meta-data including information of a corresponding target table for storing at least one target object with which the source object has a one-to-many relationship of privately owned type and information of a foreign key of the corresponding target table;

generating an insert instruction to add a value of the foreign key based on the primary key value of the source object, and to insert the at least one target object into the corresponding target table based on the mapping meta-data; and inserting the at least one target object into the database by executing the insert instruction on the database.

7. The method as claimed in claim 6, wherein the insert instruction generating step and the target object inserting step comprise steps of:

- a) building a database row representation of each target object containing target object data;
- b) adding the primary key value of the source object to the database row representation;
- c) generating the insert instruction based on the database row representation and the target object data using the mapping meta-data;
- d) writing the target object in a row of the corresponding target table by executing the insert instruction on the database; and
- e) repeating steps a) to d) for each target object using the mapping meta-data.

8. The method as claimed in claim 7, wherein the adding step adds the primary key value of the source object as a foreign key value of the target object.

9. The method as claimed in claim 6, wherein the statement generating step generates the insert instructions as a Structured Query Language (SQL) Insert statement.

10. The method as claimed in claim 6 further comprising a step of:
storing the mapping meta-data external to the source object class and the target object classes.

11. The method as claimed in claim 10, wherein the storing step stores the mapping meta-data as XML files.

12. The method as claimed in claim 6 further comprising steps of:
reading the source object from the object model; and

[illegible]

obtaining, when a source object having a primary key value is being read from a source table in the relational database, mapping meta-data including information of one or more corresponding target tables and information of one or more foreign keys of the corresponding target tables;

reading the target objects and relationships relating to the source object from the database by executing the select instruction on the database.

querying for rows in the target tables that have a foreign key value matching the primary key value of the source object by executing the select instruction on the database:

adding the target objects to a collection that represents a value of relationship of the source object, the value of relationship referencing to the target objects; and

15. The method as claimed in claim 13, wherein the generating step comprises a step of generating a select instruction to check foreign key fields in the target tables.

17. The method as claimed in claim 16 further comprising a step of:

storing the mapping meta-data external to the source object class and the target object classes.

18. The method as claimed in claim 17, wherein the storing step stores the mapping meta-data as XML files.

19. A method for managing object to relational one-to-many mapping for an object model mapped to a relational database, the method comprising steps of:

obtaining, when a source object having a primary key value is being deleted from a source table in the relational database; mapping meta-data that defines one or more corresponding target tables storing target objects with which the source object has one-to-many relationships of privately owned type and foreign key information;

generating a delete instruction to delete the target objects from the target tables based on the mapping meta-data; and

deleting the target objects by executing the delete instruction on the database.

20. The method as claimed in claim 19 further comprising a step of determining whether the target objects are stored in a single target table, and

wherein, when the target objects are determined to be stored in a single target table, the delete instruction generating step comprises a step of generating a single delete instruction that deletes all rows from the target table that have a foreign key matching the primary key of the source object.

21. The method as claimed in claim 19 further comprising a step of determining whether the target objects are stored in multiple target tables; and

wherein, when the target objects are determined to be stored in multiple target tables, the delete instruction generating step comprises steps of:

extracting a primary key value for each target object;

generating a delete instruction to delete each row with the primary key value;

and

repeating the extracting step and the generating step.

22. The method as claimed in claim 19, wherein the statement generating step generates the delete instructions as SQL Delete statements.

23. The method as claimed in claim 19 further comprising a step of:
storing the mapping meta-data external to the source object class and the target object classes.

24. The method as claimed in claim 23, wherein the storing step stores the mapping meta-data as XML files.

25. A method for managing object to relational one-to-many mapping for an object model mapped to a relational database, the method comprising a step of tracking changes to relationships and target objects with which the source object has one-to-many relationships of privately owned type, wherein the tracking step comprises steps of:

creating, at the start of a transaction, a source clone of a source object having a primary key value, a relationship clone of a relationship in the source object, and a target clone of target objects referenced by the relationship; and
determining, when the transaction is committed, changes to the source object and target objects by comparing the source clone to the current state of the source object, and comparing the target clone to the current state of the target objects using the primary key value of the source object added to the target objects.

26. The method as claimed in claim 25 further comprising a step of updating relationship of the source object referencing target objects to the database wherein the updating step comprises steps of:

determining which objects have been added, removed or changed in the relationship using mapping meta-data including information of corresponding source table, one or more target tables and foreign keys of the target tables;

generating an insert instruction for each target object of a source target that has been added;

generating a delete instruction for each target object of a source target that has been removed; and

generating an update instruction for each target object of a source target that has been changed.

a representation builder for building a database row representation for each target object; and

a primary key information adder for adding the primary key value of the source object to the database row representation.

33. The manager as claimed in claim 31, wherein the instruction generator generates the insert instructions as SQL Insert statements.

34. The manager as claimed in claim 30, wherein the instruction generator having a read instruction generator for generating read instructions to read the multiple target objects and the relationship from the corresponding target tables and the foreign key values based on the mapping meta-data when a source object is read from the database.

35. The manager as claimed in claim 34, wherein the instruction generator generates the read instructions as SQL Select statements.

36. The manager as claimed in claim 30, wherein the instruction generator having a delete instruction generator for generating delete instructions to read the multiple target objects and the relationship from the corresponding target tables and the foreign key values based on the mapping meta-data when a source object is deleted from the database.

37. The manager as claimed in claim 36, wherein the instruction generator generates the delete instructions as SQL Delete statements.

38. The manager as claimed in claim 30, wherein the instruction generator having an update instruction generator for generating update instructions to track changes to the multiple target objects and the relationship, and to update the database when a source object is changed in the database.

39. The manager as claimed in claim 38, wherein the instruction generator generates the read instructions as SQL Insert statements, SQL Delete statements or SQL Update statements depending on the change to the source object.

40. A mapping system for managing an object model to a relational database, the object model containing object classes having one or more source objects and target objects having one-to-many relationships of privately owned type with the source objects, the relational database having tables including one or more source tables for storing the source objects and one or more target tables for storing the target objects, the source tables having primary keys and the target tables having foreign keys, the system comprising:

a mapping tool for assisting in mapping object classes to tables and mapping relationships to foreign keys;

a meta-data storage for storing mapping meta-data defining information of how object classes map to tables and information of how relationships map to foreign keys, the mapping meta-data including information of one or more corresponding target tables for storing the multiple target objects and information of one or more foreign keys of the corresponding target tables; and

a runtime mapping library for accessing the mapping meta-data in the meta-data storage and managing object data in the database, the runtime mapping library having a one-to-many mapping manager for managing one-to-many relationships and target objects with which a source objects have one-to-many relationships in accordance with the manipulation of the source object.

41. The mapping system as claimed in claim 40, wherein the one-to-many mapping manager comprises:

an instruction generator for generating instructions to manage the multiple target objects and the relationship based on the mapping meta-data; and

an instruction executor for executing the instructions on the database and managing the target objects and the relationship in the database in accordance with the manipulation of the source object.

42. The manager as claimed in claim 40, wherein the meta-data storage is provided external to the source object class and the target object classes.

43. The manager as claimed in claim 40, wherein the meta-data storage is an XML file system for storing the mapping meta-data in XML files.

at least one target object with which the source object has a one-to-many relationship of privately owned and information of a foreign key of the corresponding target table;

a module for generating an instruction to manipulate the at least one target object in the corresponding target table based on the mapping meta-data; and

a module for manipulating the at least one target object in the database by executing the instruction on the database.